

## Screen 1

You are an employer and you are creating a program and the application form. 1. Use the POST method to correctly store the different types of questions we have (Paragraph, YesNo, Dropdown, MultipleChoice, Date, Number).

## Solution

The interface below shows the endpoint that would be use used to submit question structures:

The screenshot displays a REST client interface with the following details:

- Curl:** A terminal window showing the curl command used to send a POST request to `https://localhost:7180/api/questions` with a JSON body containing question details.
- Request URL:** `https://localhost:7180/api/questions`
- Server response:** A status code of 201 (Created) with a response body containing the created question's details, including an ID, paragraph text, and question type.
- Response headers:** Headers indicating the content type is application/json, the date is Wed, 19 Jun 2024 02:37:21 GMT, the location is the full URL, and the server is Kestrel.
- Responses:** A table showing a 200 Success status with no links.

```
curl -X 'POST' \
  https://localhost:7180/api/questions \
  -H 'accept: text/plain' \
  -H 'content-type: application/json' \
  -d '{
    "paragraph": "Sample Question for Yes or No category",
    "yesNo": true,
    "dropdown": false,
    "multipleChoice": false,
    "date": "2024-06-19T02:35:46.926Z",
    "number": 2
  }'
```

Request URL

https://localhost:7180/api/questions

Server response

Code Details

201  
Undocumented

Response body

```
{
  "id": "1a68b971-a250-4db8-abd6-b96535adcb5d",
  "paragraph": "Sample Question for Yes or No category",
  "yesNo": true,
  "dropdown": false,
  "multipleChoice": false,
  "date": "2024-06-19T02:35:46.926Z",
  "number": 2,
  "questionType": "YesNo"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 19 Jun 2024 02:37:21 GMT
location: https://localhost:7180/api/questions/1a68b971-a250-4db8-abd6-b96535adcb5d
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

Media type: text/plain

2. Use the PUT method if the user wants to edit the question after creating the application.

## Solution

Find the output created by the PUT method

Curl

```
curl -X 'PUT' \
  'https://localhost:7180/api/questions/1a68b971-a250-4db8-abd6-b96535adcb5d' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": "1a68b971-a250-4db8-abd6-b96535adcb5d",
    "paragraph": "Sample Question for Yes or No type of question",
    "yesNo": true,
    "dropdown": false,
    "multipleChoice": false,
    "date": "2024-06-19T00:00:00Z",
    "number": 2
  }'
```

Request URL

https://localhost:7180/api/questions/1a68b971-a250-4db8-abd6-b96535adcb5d

Server response

Code Details

200

Response body

```
{
  "id": "1a68b971-a250-4db8-abd6-b96535adcb5d",
  "paragraph": "Sample Question for Yes or No type of question",
  "yesNo": true,
  "dropdown": false,
  "multipleChoice": false,
  "date": "2024-06-19T00:00:00Z",
  "number": 2,
  "questionType": "YesNo"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 19 Jun 2024 02:40:49 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

## Screen 2

Candidate applying for the program.

1. You should provide a GET endpoint for the front-end team to render the questions based on the question type.

### Solution:

Please find the output of the endpoint for rendering questions based on type:

Curl

```
curl -X 'GET' \
  'https://localhost:7180/api/questions/search/YesNo' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7180/api/questions/search/YesNo

Server response

Code Details

200

Response body

```
{
  "id": "1a68b971-a250-4db8-abd6-b96535adcb5d",
  "paragraph": "Sample Question for Yes or No type of question",
  "yesNo": true,
  "dropDown": false,
  "multipleChoice": false,
  "date": "2024-06-19T00:00:00Z",
  "number": 2,
  "questionType": "YesNo"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 19 Jun 2024 02:47:15 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	SUCCESS	No links

Media type

text/plain

Controls Accept header:

Evamnia Value | Schema

2. Provide a POST endpoint for the front-end team with the payload structure so that once the candidate submits the application, you will receive the data.

### Solution:

Please find the output of the endpoint for receiving candidates' applications:

Curl

```
curl -X 'POST' \
  'https://localhost:7180/api/candidates' \
  -H 'accept: text/plain' \
  -H 'content-type: application/json' \
  -d '{
    "firstName": "John",
    "lastName": "Doe",
    "email": "testemail@gmail.com",
    "phone": "+199988878",
    "nationality": "USA",
    "currentResidence": "Caliphonia",
    "idNumber": "676555",
    "dateOfBirth": "2024-06-19T03:50:07.107Z"
  }'
```

Request URL

https://localhost:7180/api/candidates

Server response

Code Details

201

Undocumented

Response body

```
{
  "id": "6aebbf4f-47dc-48e9-aa3e-6d4660e2b9f8",
  "firstName": "John",
  "lastName": "Doe",
  "email": "testemail@gmail.com",
  "phone": "+199988878",
  "nationality": "USA",
  "currentResidence": "Caliphonia",
  "idNumber": "676555",
  "dateOfBirth": "2024-06-19T03:50:07.107Z",
  "category": "Basic"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 19 Jun 2024 03:51:34 GMT
location: https://localhost:7180/api/candidates/6aebbf4f-47dc-48e9-aa3e-6d4660e2b9f8
server: Kestrel
```

Responses

Code	Description	Links
------	-------------	-------

The implementation can be expanded later to contain questions for the candidates.

**I actually saw the figma design very late and that really affected my work. I saw this line in the document "Create DTOs & Models based on the Figma designs below" and I kept checking below. I could have done a better job in this task. Thank you for the opportunity all the same.**